

IN THE CLAIMS:

Please amend claims 1-21 and add new claim 22 as follows:

1. (currently amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for implementing an object type-declaration syntax, comprising:
- allowing a type declaration in a programming language to be embedded within an object identifier declaration; and
 - allowing the type declaration to be delimited from the object identifier declaration using a joint attribute, the joint attribute used by an interpreter or a compiler of the programming language to separate the type declaration from the object identifier to determine an object type of an object being declared in the object identifier when the object identifier having the type declaration is read, the determining done without requiring additional referencing each time the object identifier having the type declaration is read; and
 - allowing the object identifier with the embedded type declaration to be used throughout a life of a program as a syntax for referencing an object in the program.
2. (currently amended) The program storage device of claim 1, wherein the allowing a type declaration includes allowing a type declaration in a programming language compiler to be embedded within an object identifier declaration.
3. (previously amended) The program storage device of claim 1, wherein the type declaration includes a database object type.
4. (previously amended) The program storage device of claim 1, wherein the type declaration includes a SQL database object type.
5. (previously amended) The program storage device of claim 1, wherein the type declaration includes a connection database object type.
6. (previously amended) The program storage device of claim 1, wherein the type declaration includes a cursor database object type.

7. (previously amended) The program storage device of claim 1, wherein the type declaration includes a universal resource locator object type.

8. (previously amended) The program storage device of claim 1, wherein the type declaration includes an environment object type.

9. (previously amended) The program storage device of claim 1, wherein the type declaration includes a hypertext markup language object type.

10. (previously amended) The program storage device of claim 1, wherein the type declaration includes an extensible markup language object type.

11. (previously amended) The program storage device of claim 1, wherein the joint attribute is concatenated to the type declaration.

12. (currently amended) The program storage device of claim 11, wherein the object identifier ~~declaration~~ is concatenated to the joint attribute.

13. (currently amended) The program storage device of claim 1, wherein the joint attribute is concatenated to the object identifier ~~declaration~~.

14. (previously amended) The program storage device of claim 13, wherein the type declaration is concatenated to the joint attribute.

15. (currently amended) The object type-declaration syntax as claimed in claim 1, wherein the object identifier ~~declaration~~ includes dynamically evaluated expressions.

16. (currently amended) A method of declaring an object type in a programming language, comprising:

embedding an object type indicator with an object identifier name, the object type indicator and the object identifier name delineated by a predefined symbol, wherein the object

~~identifier name is interpreted by a machine~~ uses the predefined symbol to separate the object type indicator and the object identifier name to identify an object type for the object identifier name and as having the object type indicator the object identifier name having the object type indicator is used throughout a life of a program as a syntax for referencing an object in the program.

17. (currently amended) A method of declaring an object type in a programming language, comprising:

By prepending an object type ~~indieater~~ indieater followed by a predefined symbol to ~~with an object identifier string name, the object type, the predefined symbol, and the object identifier string forming a symbol name to be carried throughout a life of a program as a syntax for referencing an object in the program, wherein the object identifier name is interpreted by a machine interpreting the symbol name in the program uses the predefined symbol to delineate the object type from the object to determine the object type as having the object type indieater.~~

18. (original) The method of declaring an object type in a programming language as claimed in claim 16, wherein the step of embedding includes:

joining the object type indicator with the object identifier name with a joint symbol.

19. (canceled).

20. (currently amended) The program storage device of claim 1, wherein the type declaration allows a compiler or interpreter of the programming language to operate on an object declared in the type declaration without an explicit call to construct the object.

21. (currently amended) The program storage device of claim 1, wherein the type declaration allows a compiler or interpreter of the programming language to automatically instantiate an object being declared in the type declaration when the type declaration embedded with the object identifier ~~declaration~~ is first read by the programming language compiler or interpreter.

22. (new) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method of implementing a program syntax for declaring an object type and using an object in a program, comprising:

Ad. Gov.

integrating an explicit object type definition within a string of characters that define an object symbol name that is used throughout a program for referencing an object, the object type definition delineated by an additional predefined symbol in the string of characters, the additional predefined symbol being an explicit symbol separate from the explicit object type definition, wherein a machine interpreting the object symbol name throughout the program determines what type the object symbol name is by recognizing the additional predefined symbol in the string of characters and reading the explicit object type definition in the string of characters.
